# Documentation for src\main\java\com\example\EmployeeManagementSystem\service\ServiceEmployee.java

Generated on: 2025-09-05 22:42:26

Repository: C:\Users\User\VisualStudio\Employee-Management-Sys\EmployeeManagementSystem

Language: java

---

# createEmployee Function

## Description

The createEmployee function is used to add a new employee to the Employee Management System. It takes a dtoEmployee object as a parameter and creates a new employee record in the system.

## Parameters

* dtoEmployee: This parameter is a data transfer object that contains the information of the employee to be added.

## Usage Example

DtoEmployee newEmployee = new DtoEmployee("John Doe", "john.doe@example.com", "Manager");

service.createEmployee(newEmployee);

## Important Notes

* Make sure to provide all the necessary information in the dtoEmployee object before calling the `createEmployee` function.
* This function does not return any value, it simply adds the employee to the system.\n\n---\n\n# Function: findById

## Description

This function retrieves an employee record from the database based on the provided id.

## Parameters

* id: The unique identifier of the employee to be retrieved.

## Usage Example

Employee employee = serviceEmployee.findById(123);

System.out.println(employee);

## Important Notes

* Ensure that the id provided is valid and corresponds to an existing employee record in the database.
* This function returns null if no employee is found with the provided id.\n\n---\n\n# findAllEmployee Function Documentation

## Description

This function retrieves all employees from the Employee Management System.

## Parameters

None

## Usage Example

List<Employee> employees = findAllEmployee();

for(Employee employee : employees) {

System.out.println(employee.getName());

}

## Important Notes

* This function does not require any parameters.
* Make sure to handle any potential exceptions that may occur when calling this function.\n\n---\n\n# updateEmployee Function

## Description

This function updates an employee's information in the Employee Management System.

## Parameters

* `id`: The unique identifier of the employee to be updated.
* `updatedemployee`: The updated information of the employee.

## Usage Example

// Update employee with id 123

Employee updatedEmployee = new Employee("John Doe", "john.doe@example.com", "Manager");

service.updateEmployee(123, updatedEmployee);

## Important Notes

* Make sure to provide the correct `id` of the employee to be updated.
* The `updatedemployee` parameter should contain the new information for the employee.\n\n---\n\n# Function: deleteEmployee

## Description

This function deletes an employee from the employee management system based on the provided employee ID.

## Parameters

* id: The unique identifier of the employee to be deleted.

## Usage Example

ServiceEmployee serviceEmployee = new ServiceEmployee();

serviceEmployee.deleteEmployee(12345);

## Important Notes

* Make sure to provide a valid employee ID to delete the correct employee from the system.
* This action cannot be undone, so double-check before deleting an employee.\n\n---\n\n